PUC Minas

Algoritmos e Técnicas de Programação

Edwaldo Soares Rodrigues
Departamento de Ciência da Computação / Sistemas de Informação
PUC Minas São Gabriel
2023/1

Estruturas de Dados Homogêneas - Vetores

- Motivação:
  - Exercício: Faça um programa que leia n números inteiros, calcule a média desses valores e armazene e mostre aqueles que forem maiores que a média.

```
1   using System;
2   class media
3   {
4       static void Main(string[] args)
5       {
6           int n; double valor, media = 0;
7           Console.Write("Entre com o valor de n: ");
8           n = int.Parse(Console.ReadLine());
9           for(int i=0; i < n; i ++){
10              Console.WriteLine("Digite o {0} valor: ", i+1);
11              valor = double.Parse(Console.ReadLine());
12              media += valor;
13          }
14          media = media/n;
15          Console.WriteLine("\n\n A media e igual a: " + media);
16      }
17  }
```

Como mostrar os elementos maiores do que a média?

PUC Minas

Estruturas de Dados Homogêneas - Vetores

- Motivação:
  - Pelo que vimos até agora, cada variável identifica um dado da memória;
  - Porém, sabemos que programas complexos trabalham com muitos dados: milhares de estudantes, milhões de produtos, ou centenas de números;
  - Precisamos então de estruturas que representem vários dados de uma só vez;

Estruturas de Dados Homogêneas - Vetores

- Vetores (Arrays – Arranjos):
  - Um arranjo é um endereço onde há um grupo consecutivo de posições alocadas na memória;
  - Todas as posições neste grupo guardam variáveis do mesmo tipo;
  - Arranjos são estruturas de dados muito úteis para se trabalhar com vários itens do mesmo tipo;

Estruturas de Dados Homogêneas - Vetores

- Benefícios da utilização de Vetores:
  - Ajudam a manter uma coleção de dados;
  - Permitem manter as informações organizadas;
  - Permitem operações com o volume de dados neles inseridos;
  - Antes tínhamos n variáveis pra guardar n valores;
  - Agora teremos uma variável para armazenar n valores;

Estruturas de Dados Homogêneas - Vetores

- Características dos Vetores:
  - Conhecidos em C# como arrays (arranjos);
  - Correspondem a posições de memória;
  - São identificados por um nome;
  - Individualizados por índices;
  - Conteúdo do mesmo tipo;
  - Podem ser classificados em unidimensionais ou multidimensionais;

Estruturas de Dados Homogêneas - Vetores

- Indexação:
  - Resumindo: Vetores são posições de memória identificadas por um mesmo nome, individualizadas por índices e cujo conteúdo é do mesmo tipo.
  - Exemplo:

Este é um arranjo de números inteiros chamado x.

PUC Minas

Estruturas de Dados Homogêneas - Vetores

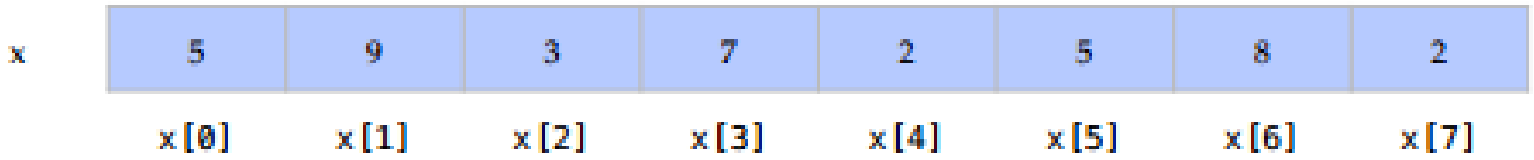
- Indexação:
  - Resumindo: Vetores são posições de memória identificadas por um mesmo nome, individualizadas por índices e cujo conteúdo é do mesmo tipo.
  - Exemplo:

X nada mais é que o endereço do primeiro elemento deste arranjo de números inteiros.

PUC Minas

Estruturas de Dados Homogêneas - Vetores

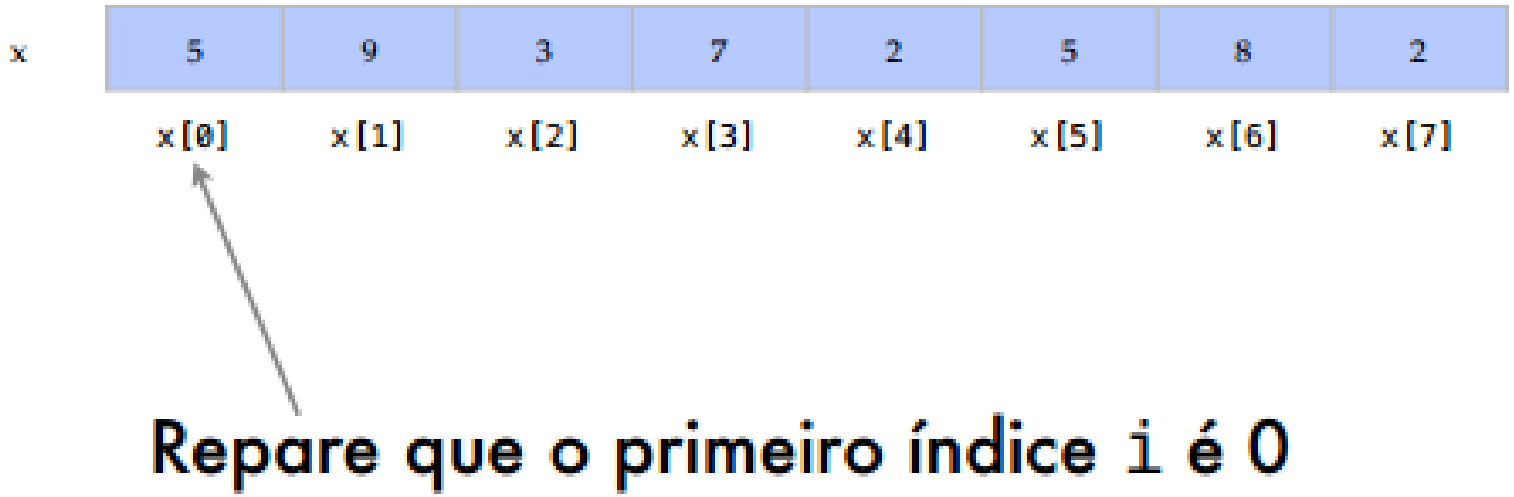
- Acesso às posições:
  - Para acessar elementos específicos precisamos utilizar x[i] onde i é a posição do arranjo. Formalmente i é chamado de índice do arranjo:



PUC Minas

Estruturas de Dados Homogêneas - Vetores

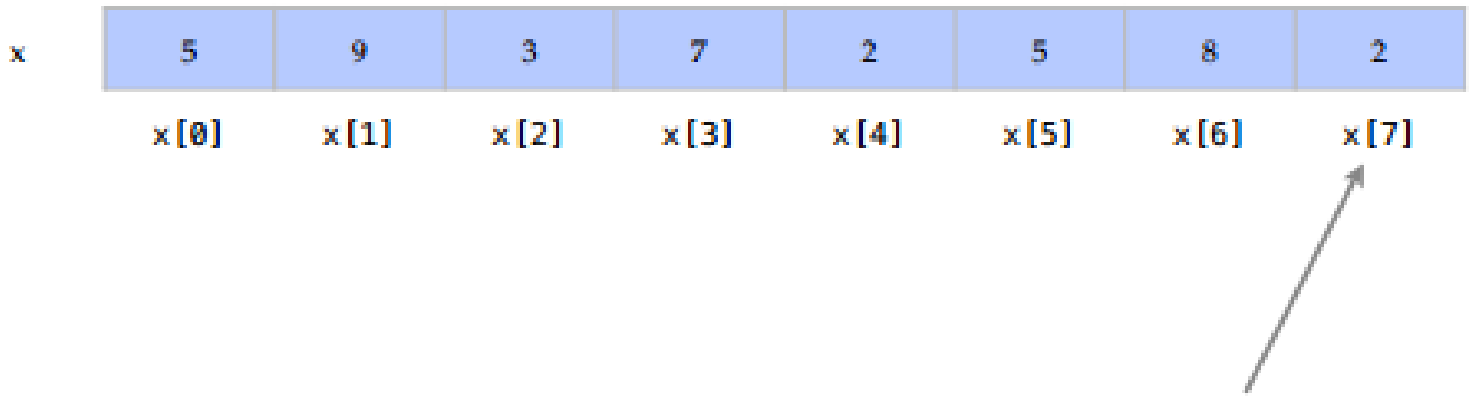
- Acesso às posições:
  - Para acessar elementos específicos precisamos utilizar x[i] onde i é a posição do arranjo. Formalmente i é chamado de índice do arranjo:



PUC Minas

Estruturas de Dados Homogêneas - Vetores

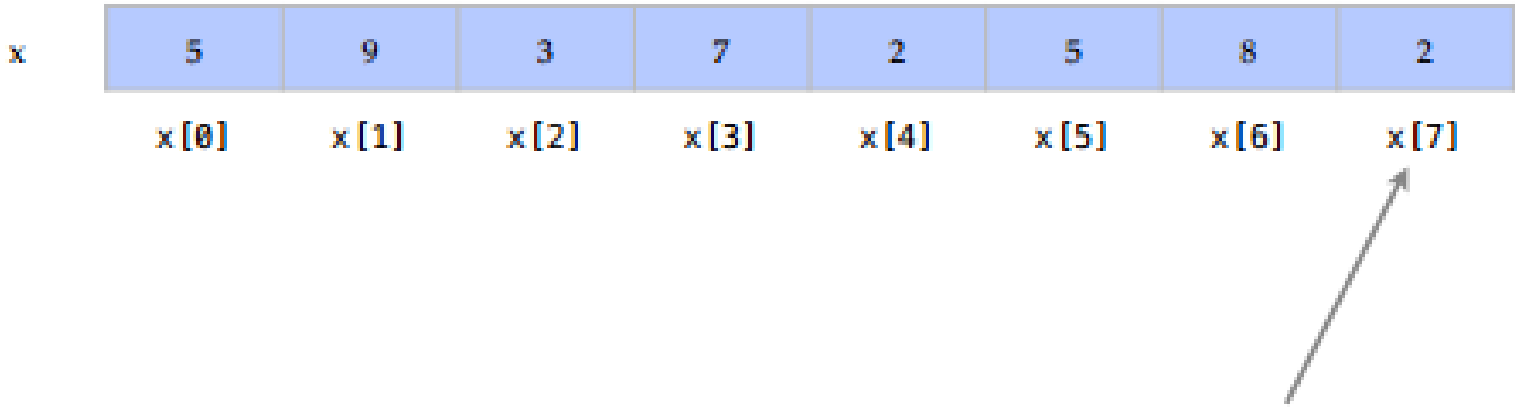
- Acesso às posições:
  - Para acessar elementos específicos precisamos utilizar x[i] onde i é a posição do arranjo. Formalmente i é chamado de índice do arranjo:



PUC Minas

Estruturas de Dados Homogêneas - Vetores

- Acesso às posições:
  - Para acessar elementos específicos precisamos utilizar x[i] onde i é a posição do arranjo. Formalmente i é chamado de índice do arranjo:

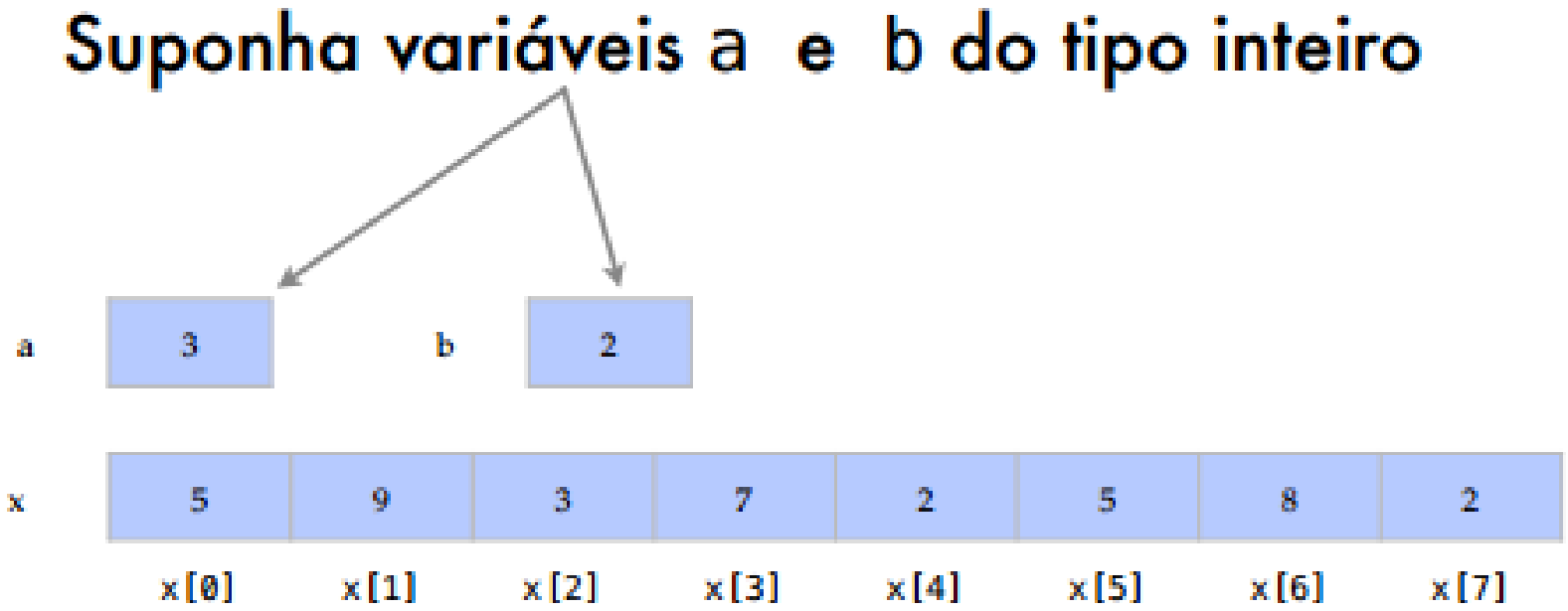


Os índices são simplesmente números inteiros ou uma expressão do tipo inteiro

PUC Minas

Estruturas de Dados Homogêneas - Vetores

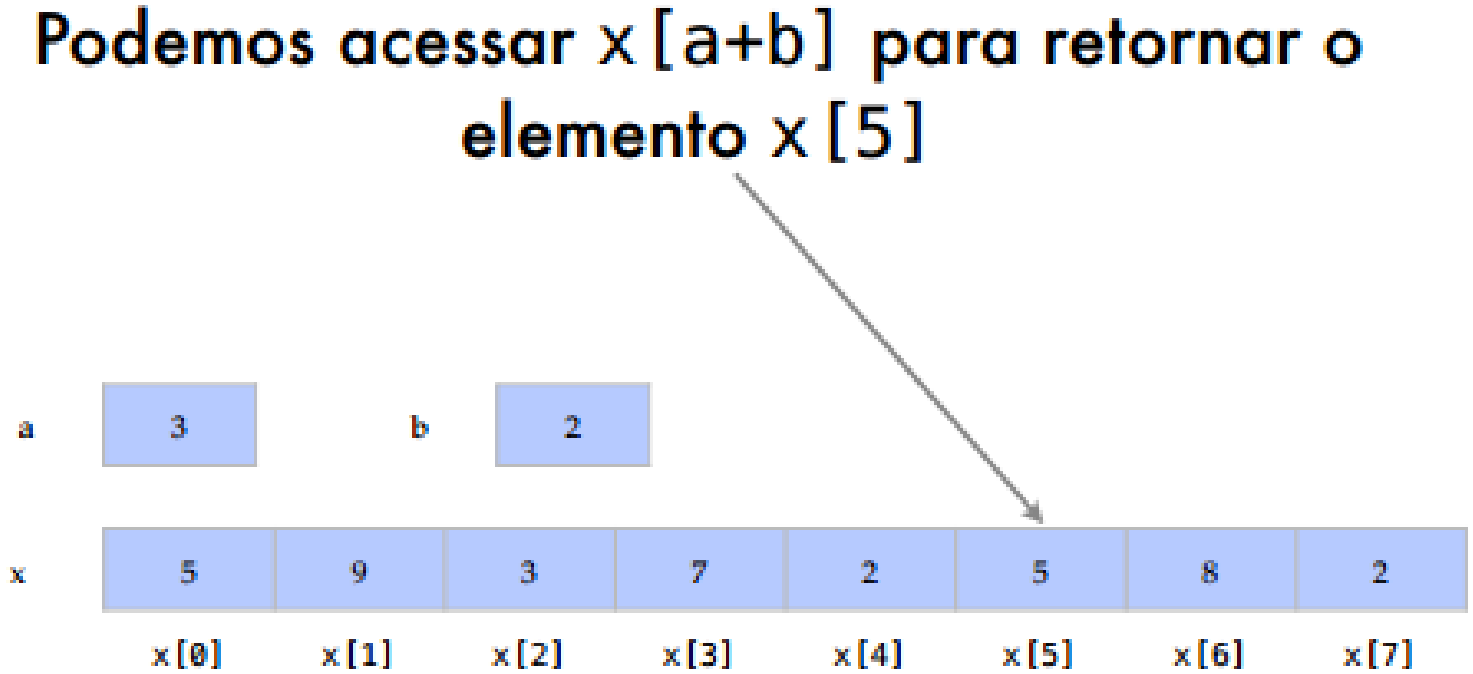
- Acesso às posições:



PUC Minas

Estruturas de Dados Homogêneas - Vetores

- Acesso às posições:



PUC Minas

Estruturas de Dados Homogêneas - Vetores

- Acesso às posições:

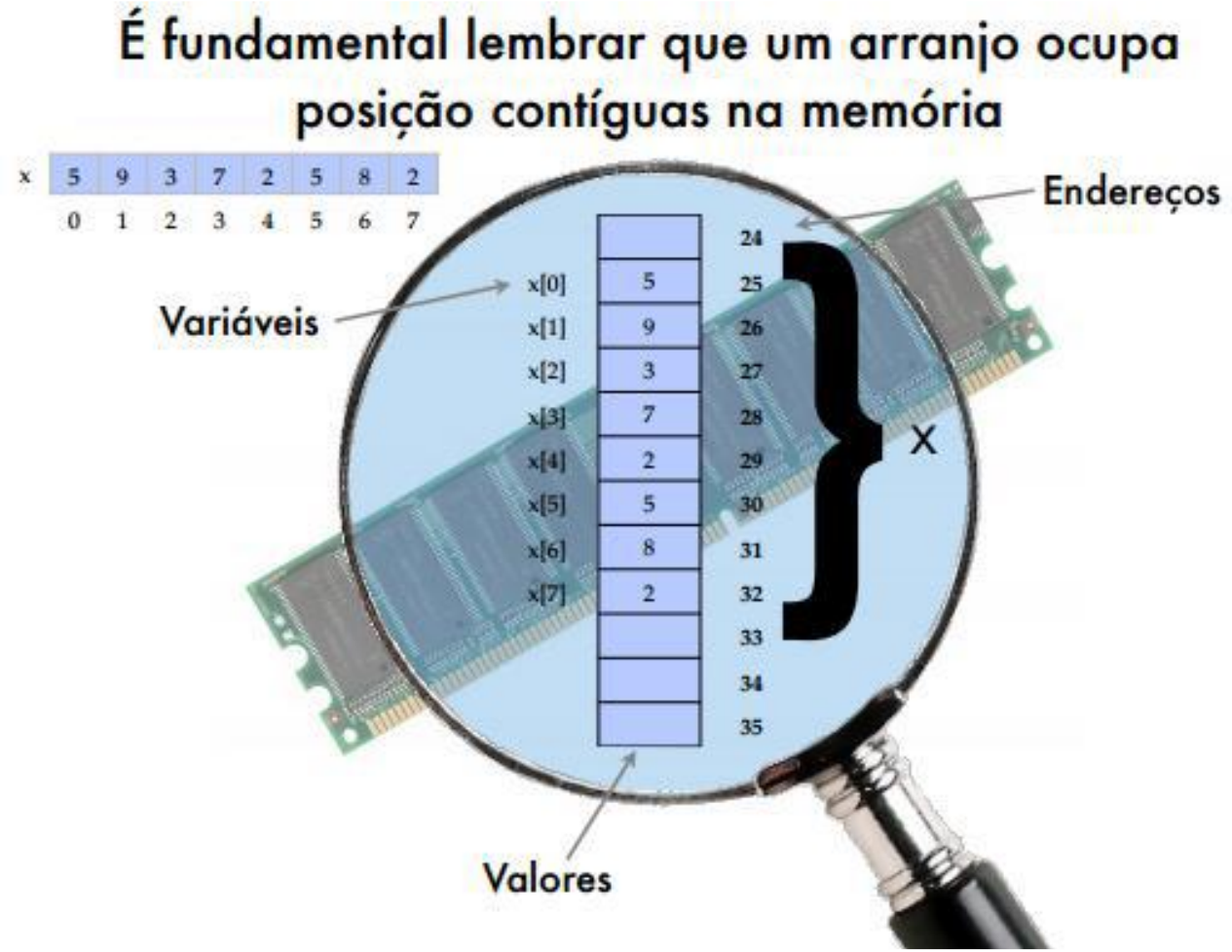
Os endereços dos elementos do arranjo estão em sequência na memória.

Endereço	50	51	52	53	54	55	56	57
x	5	9	3	7	2	5	8	2
	x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]

PUC Minas

Estruturas de Dados Homogêneas - Vetores

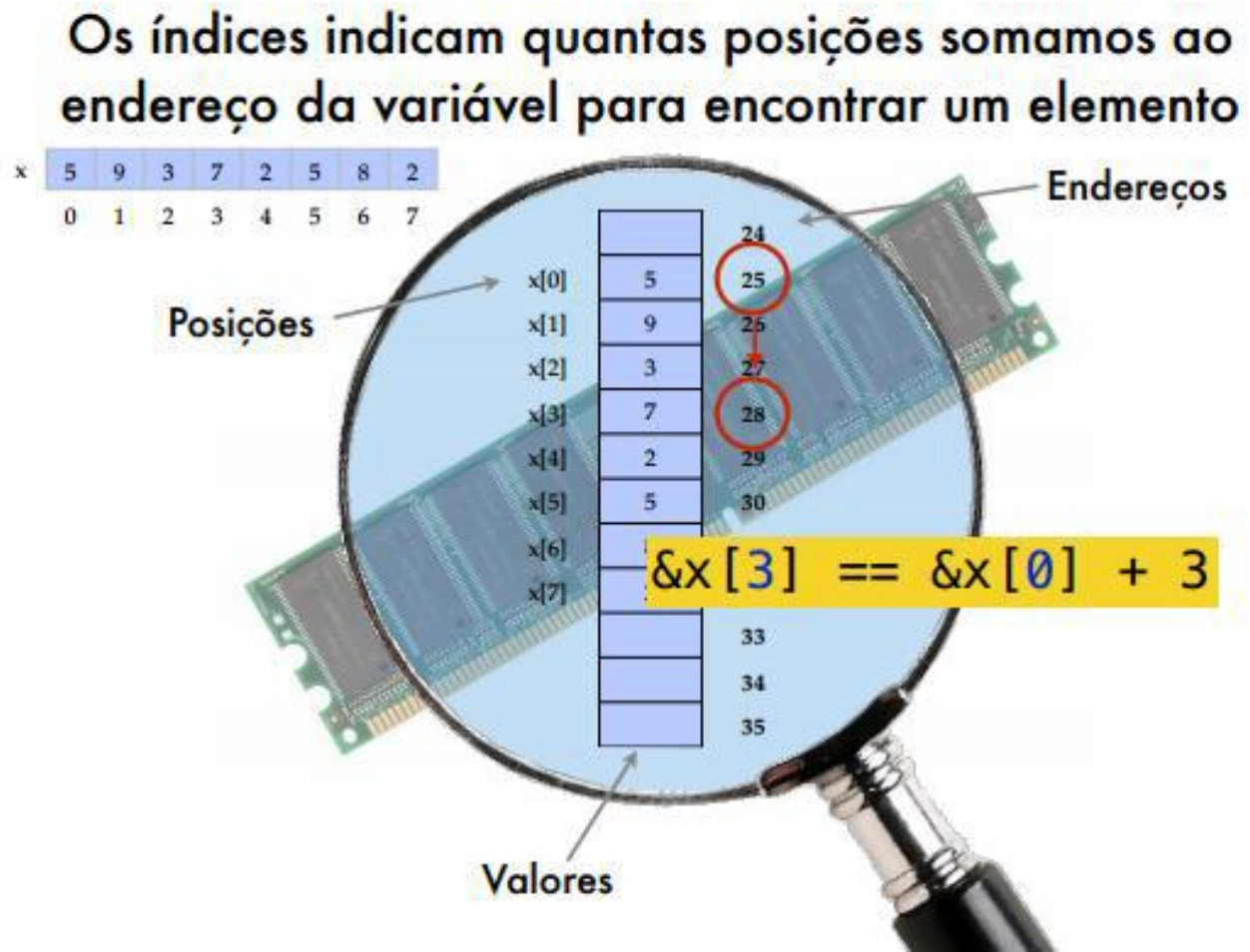
- Acesso às posições:



PUC Minas

Estruturas de Dados Homogêneas - Vetores

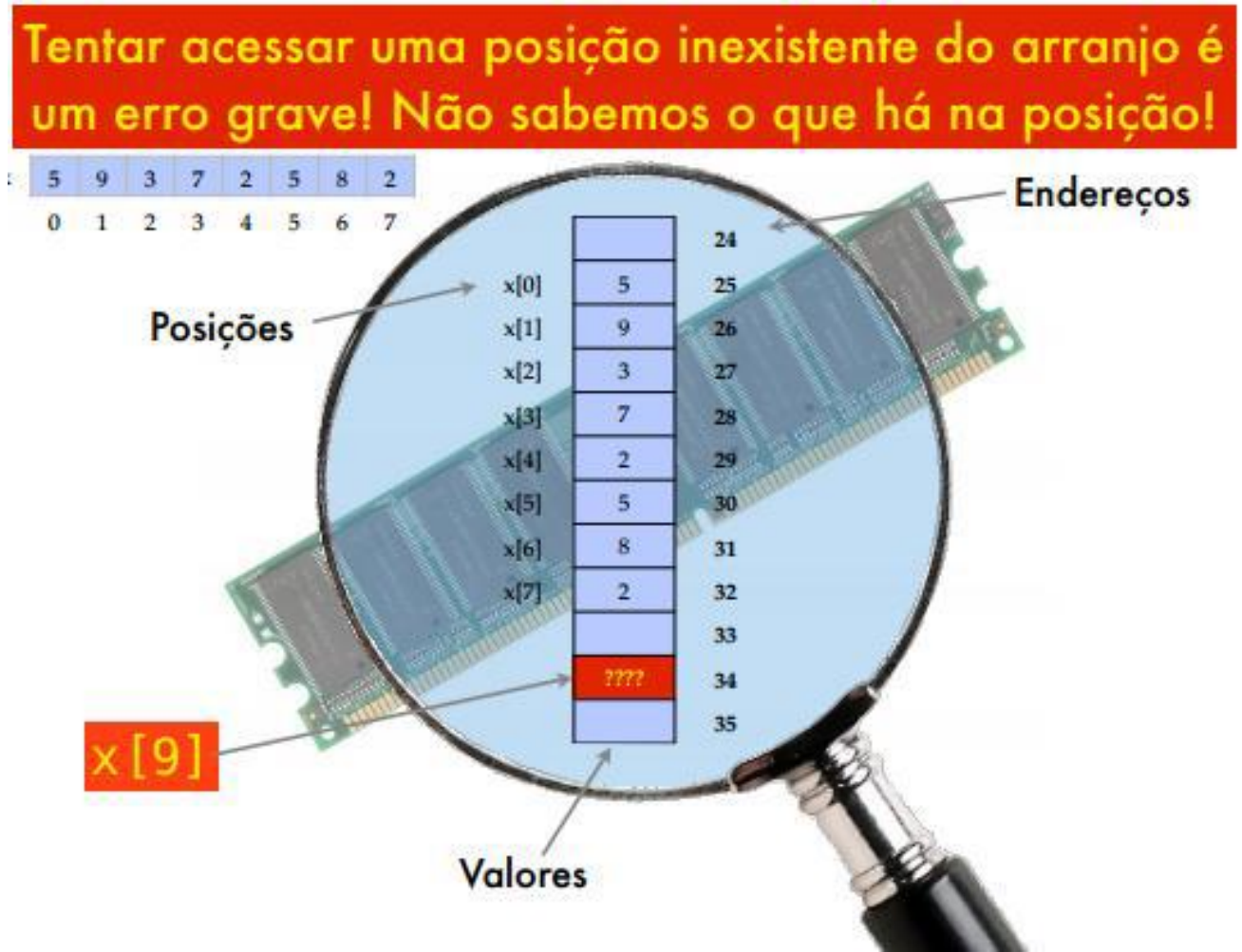
- Acesso às posições:



PUC Minas

Estruturas de Dados Homogêneas - Vetores

- Acesso às posições:



PUC Minas

Estruturas de Dados Homogêneas - Vetores

- Criação de um vetor:
  - Os arrays são objetos; portanto, são considerados tipos por referência;
  - Em C# quando um arranjo é criado, cada elemento recebe um valor padrão — zero para elementos de tipo primitivo numéricos, false para elementos booleanos;

Estruturas de Dados Homogêneas - Vetores

- Criação de um vetor - Sintaxe:
  - tipo[ ] nome_array = new tipo [tamanho];
  - Exemplos:
    - 1) int [ ] vet1 = new int [10];
    - 2) int [ ] vet2;
      vet2 = new int [10];
    - 3) int [ ] vet3 = {5, 6, 20, 40, 2, 34, 87, 3, 1, 4};

O valor inicial de vet1 e vet2 será zero e de vet3 será os valores passados entre {}

Ao inicializar uma variável de array, pode-se omitir a expressão new e o tamanho do array. O compilador calcula o tamanho a partir do número de inicializadores.

Estruturas de Dados Homogêneas - Vetores

- Acessando posições inválidas:
  - Se um vetor tem tamanho n, as posições válidas são de 0 a (n-1);
  - Tentar acessar posições negativas ou maiores que (n-1) ocasionam erros no programa;

```
int n = 10;
int [] vet1 = new int[n];
            ...
for(int i = n - 1; i >= 0; i--)
{
    vet1[i-1] = vet1[i];
}
```

```
int n = 10;
int [] vet1 = new int[n];
            ...
for(int i = n; i >= 0; i--)
{
    escreva: vet1[i];
}
```

PUC Minas

Estruturas de Dados Homogêneas - Vetores

- Tamanho de um vetor:
  - Um objeto array conhece seu comprimento e armazena essas informações em uma variável de instância ***length***.
- Como usar:
  - Nome-do-vetor.Length
- Exemplo:

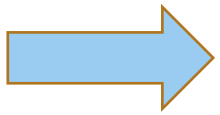
```
for (int i =0; i < valores.Length; i++)
{
    .... ;
}
```

PUC Minas

Estruturas de Dados Homogêneas - Vetores

- Laço foreach:
  - Percorre o array inteiro automaticamente, obtendo um elemento de cada vez, em sequência, do início ao fim. Percorre do índice menor para o maior;

```
int [ ] nums = {1,2,3,4,5,6,7};
int soma = 0;
for (int i = 0; i<7; i++)
    soma += nums [i];
```



```
int [ ] nums = {1,2,3,4,5,6,7};
int soma = 0;
foreach(int n in nums){
    soma+=n;
  }
```

  - A variável declarada no foreach é de somente “leitura”, ou seja, não podemos atribuir um valor a ela explicitamente.

PUC Minas

Estruturas de Dados Homogêneas - Vetores

- Laço foreach:
  - O foreach é um recurso do C# que possibilita executar um conjunto de comandos para cada elemento presente em uma coleção (Array, List, Stack, Queue e outras);
  - Portanto, diferentemente do while, do do-while e do for, não precisamos definir uma condição de parada. Isso é definido de forma implícita, pelo tamanho da coleção;

Estruturas de Dados Homogêneas - Vetores

- Motivação:
  - Exercício: Faça um programa que leia n números inteiros, calcule a média desses valores e mostre aqueles que forem maiores que a média.

```
1   using System;
2   class media
3   {
4       static void Main(string[] args)
5       {
6           int n; double valor, media = 0;
7           Console.Write("Entre com o valor de n: ");
8           n = int.Parse(Console.ReadLine());
9           for(int i=0; i < n; i ++){
10              Console.WriteLine("Digite o {0} valor: ", i+1);
11              valor = double.Parse(Console.ReadLine());
12              media += valor;
13          }
14          media = media/n;
15          Console.WriteLine("\n\n A media e igual a: " + media);
16      }
17  }
```

Como mostrar os elementos maiores do que a média?

PUC Minas

Estruturas de Dados Homogêneas - Vetores

- Motivação:

```
1   using System;
2   class media
3   {
4       static void Main(string[] args)
5       {
6           int n, cont=0; double media = 0;
7           Console.Write("Entre com o valor de n: ");
8           n = int.Parse(Console.ReadLine());
9           double[] valor = new double[n];
10
11          for(int i=0; i < n; i ++){
12              Console.WriteLine("Digite o {0} valor: ", i+1);
13              valor[i] = double.Parse(Console.ReadLine());
14              media += valor[i];
15          }
16          media = media/n;
17          Console.WriteLine("\n\nA media e igual a: " + media);
18          for(int i=0; i < valor.Length; i++){
19            if(valor[i] > media){
20              cont++;
21            }
22          }
```

PUC Minas

Estruturas de Dados Homogêneas - Vetores

- Motivação:

```
23          double[] maiores = new double[cont];
24          for(int i=0, j=0; i < valor.Length; i++){
25            if(valor[i] > media){
26              maiores[j] = valor[i];
27              j++;
28            }
29          }
30          Console.WriteLine("\n\nValores maiores que a media:\n ");
31          for(int i=0; i < maiores.Length; i++){
32            Console.WriteLine("{0} ", maiores[i]);
33          }
34      }
35  }
```

PUC Minas

Estruturas de Dados Homogêneas - Vetores

- Exercícios:
  - Faça um programa que leia 5 valores reais e armazene-os em um vetor. Na sequência seu programa deverá imprimi-los e posteriormente, calcular e imprimir a soma dos valores lidos.
  - Faça um programa que leia as notas finais de 10 alunos. Em seguida, seu programa deverá imprimir a nota do aluno que obteve a maior nota.
  - Faça um programa que leia 10 números inteiros do teclado e os imprima na ordem inversa da leitura.

Estruturas de Dados Homogêneas - Vetores

- Exercícios:

```
1   using System;
2   class media
3   {
4       static void Main(string[] args)
5       {
6         double[] v = new double[10];
7         double soma = 0;
8         int i;
9         Console.WriteLine("\n\nDigite 10 valores: \n");
10        for(i = 0; i < 10; i++){
11          Console.Write("Digite o {0}º valor: ", i+1);
12          v[i] = double.Parse(Console.ReadLine());
13          soma += v[i];
14        }
15        Console.WriteLine("\n\nOs numeros lidos foram: ");
16        foreach(double n in v){
17          Console.Write(n + " ");
18        }
19        Console.WriteLine("\n\nSoma dos numeros = " + soma);
20      }
21  }
```

Estruturas de Dados Homogêneas - Vetores

- Exercícios:

```
1   using System;
2   class media
3   {
4       static void Main(string[] args)
5       {
6         double[] notas = new double[10];
7         double maior = 0;
8         int i;
9         Console.WriteLine("\n\nDigite as notas dos 10 alunos: \n");
10        for(i = 0; i < 10; i++){
11          Console.Write("Digite a nota do {0}º aluno: ", i+1);
12          notas[i] = double.Parse(Console.ReadLine());
13          if(i == 0){
14            maior = notas[i];
15          }
16          if(notas[i] > maior){
17            maior = notas[i];
18          }
19        }
20        Console.WriteLine("\n\nA maior nota foi: " + maior);
21      }
22  }
```

PUC Minas

Estruturas de Dados Homogêneas - Vetores

- Exercícios:

```
1   using System;
2   class media
3   {
4       static void Main(string[] args)
5       {
6         int[] valores = new int[10];
7         int i;
8         Console.WriteLine("Entre com 10 valores: \n\n");
9         for(i = 0; i < 10; i++){
10          Console.Write("Digite o {0}º valor: ", i+1);
11          valores[i] = int.Parse(Console.ReadLine());
12        }
13        Console.WriteLine("\n\nNumeros impressos na ordem invertida: \n\n");
14        for(i = (valores.Length) -1; i >=0; i--){
15          Console.Write(valores[i] + " ");
16        }
17      }
18  }
```

PUC Minas

Referências

- Adaptado do material do prof. Alan Robert Rezende de Freitas;
- Adaptado do material do prof. André Backes;
- Adaptado do material da profª. Michelle Nery Nascimento;
- Adaptado do material do prof. Puca Huachi Vaz Penna;
- ASCENCIO et al. Fundamentos da programação de computadores: algoritmos, Pascal e C/C++ e Java. 3.ed. São Paulo: Pearson Prentice Hall, 2008.

PUC Minas